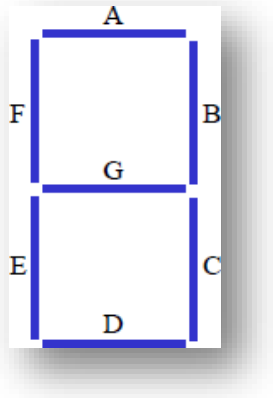
ELEC 2220 Computer Systems
Homework #13
Due: Wednesday, June 27

PART 1 – PROGRAMMING (Subroutine)

This program is to be downloaded to the flash memory and executed on the microcontroller on the STM32F4-Discovery board. The project is to include the microcontroller "startup code". The original (default) memory map of the microcontroller should be used (i.e. ROM beginning at 0x08000000 and RAM at 0x20000000). Refer to the tutorial document `Creating Projects for STM32F4-Discovery with MDK-ARM` on the course web page for information on setting up the project to run on the Discovery board.

Shown below is the format of a 7-segment display digit. Each "segment" is activated by applying a 1 to it and deactivated by applying a 0. Numbers and letters are created by activating sets of segments. For example, the digit 0 would be display by activating segments A-B-C-D-E-F. In an embedded system, 7-segment displays are often connected to parallel output ports, containing multiple pins, with one output pin connected to each display segment. An 8-bit pattern is written to each port to control the pins and therefore the segments of the attached digit. Within the 8-bit pattern, segment A corresponds to bit 0, segment B to bit 1, etc. with segment G corresponding to bit 6. Bit 7 should be set to 0.



In this exercise, an 8-digit packed BCD number is to be displayed on eight 7-segment display digits.

First, write a subroutine that will convert one BCD digit to the corresponding 7-segment code, with the bits arranged as described above. The BCD digit is to be passed as a value to the subroutine **on the stack**, and the 7-segment code should be **returned in a register**.

The main program should implement a loop, to be executed 8 times, calling the above subroutine to convert the 8 digits of a packed BCD number (which is stored in a 32-bit variable in memory) into their respective 7-segment codes, and store the codes in an array of eight bytes, each byte corresponding to one of the 8 BCD digits of the number, with the least significant digit in the first byte and the most significant digit in the last byte of the array.

For test data, generate the 7-segment codes for the packed BCD number 0x01234567, and then for the packed BCD number 0x98765432.

The data area should include:

```
Packed1     dcd    0x01234567  ;the packed BCD value to convert
SevenSeg1   space  8           ;the eight 7-segment codes
Packed2     dcd    0x98765432  ;the packed BCD value to convert
SevenSeg2   space  8           ;the eight 7-segment codes
```

Submit your source program and a copy of the debugger memory window showing the results. Annotate the window to show how each byte indicates the 7-segment code of the corresponding digit.

Also, answer the following questions.

1. At what flash memory address does the startup code begin?
2. At what flash address is the first executable instruction of your main program?
3. What RAM address corresponds to each of the above four labels in the data area?
4. What is the range of addresses allocated for the stack?

(You may be able to determine these from the listing and/or map files produced by the assembler.)

PART 2 – MICROCONTROLLER HARDWARE AND INPUT/OUTPUT PORTS

For this assignment, provide answers to the following questions. The required information can be found in the *ARM Cortex-M4 User Guide*, the *STM32F4xx Microcontroller Technical Reference*, and the *STM32F4-Discovery User Manual*. (All available from the course web page.)

1. List the range of memory addresses for the STM32F407 microcontroller's
    a. Flash memory
    b. SRAM
    c. STM microcontroller-specific peripherals
    d. Cortex-M4 peripherals
2. What is the base address of the block of registers that comprise the Reset and Clock Control (RCC) block?
3. What RCC register contains bits to enable clocks to drive the GPIO ports, and what is the address offset of this register within the RCC block?
4. What are the base addresses of the blocks of registers associated with
    a. GPIOA
    b. GPIOB
5. What are the register address offsets within the GPIO blocks for the following GPIO registers:
    a. GPIOx_MODER
    b. GPIOx_ODR
    c. GPIOx_IDR
    d. GPIOx_BSRR
6. To which GPIO pins are the four LEDs and the user button connected on the STM32F4-Discovery board?
7. Using the information from questions 2-6, write a sequence of assembly language instructions that would do the following.

    (Create and assemble the program to remove syntax errors, but do not execute it.)

    a. Turn on (enable) clocks to drive the two GPIO ports connected to the LEDs and button.
    b. Configure the GPIO pin connected to the Green LED as an output.
    c. Write a 1 to the pin to turn on the Green LED by writing to the output data register (ODR) of that GPIO port.
    d. Write a 0 to the pin to turn off the Green LED by writing to the corresponding bit set/reset register (BSSR).
    e. Configure the GPIO pin connected to the push button as an input.
    f. Read the state of the button by reading the input data register (IDR) of that GPIO port.
    g. Repeat the operation(s) in the last step until the button value is 0.